

Chapter 1

Accountability and Access Control

THE CISSP EXAM TOPICS COVERED IN THIS CHAPTER INCLUDE:

✓ Access Control

- Control access by applying the following concepts/ methodology/techniques:
 - Policies; types of controls (e.g., preventative, detective, corrective); techniques (e.g., nondiscretionary, discretionary and mandatory); identification and authentication; decentralized/distributed access control techniques; authorization mechanisms; logging and monitoring





The Access Control domain in the Common Body of Knowledge (CBK) for the CISSP certification exam deals with topics and issues related to monitoring, identifying, and authorizing or restricting user access to resources. Generally, an *access control* is any hardware, software, or organizational administrative policy or procedure that grants or restricts access, monitors and records attempts to access, identifies users attempting to access, and determines whether access is authorized.

In this chapter and in Chapter 2, “Attacks and Monitoring,” we discuss the Access Control domain. Be sure to read and study the materials from both chapters to ensure complete coverage of the essential material for this domain of the CISSP certification exam. We’ve called this chapter “Accountability and Access Control” because accountability and access control are interrelated concepts and share overlapping principles even though CISSP course materials reference only access control.

Access Control Overview

Controlling access to resources is one of the central themes of security. Access control addresses more than just controlling which users can access which files or services. Access control is about the relationships between subjects and objects. The transfer of information from an object to a subject is called *access*. However, access is not just a logical or technical concept: Don’t forget about the physical realm where access can involve disclosure, use, or proximity. A basic principle of access control is to deny access by default if access is not granted explicitly to a subject.

Subjects are active entities that, through the exercise of access, seek information about or data from passive entities, or objects. A *subject* can be a user, program, process, file, computer, database, and so on. An *object* can be a file, database, computer, program, process, file, printer, storage media, and so on. The subject is always the *entity* that receives information about or data from the object. The subject is also the entity that alters information about or data stored within the object. The object is always the entity that provides or hosts information or data. The roles of subject and object can switch back and forth while two entities, such as a program and a database or a process and a file, interact to accomplish a task. For example, when a program interacts with a database, the program begins as the subject and the database as the object when the program passes a query to the database. But when the database posts a reply to the program, the roles reverse because the database generates data that it returns to the program.

Types of Access Control

Access controls are necessary to protect the *confidentiality*, *integrity*, and *availability* of objects (and by extension, their information and data). Taken together, these three essential security principles are known as the CIA Triad. Confidentiality addresses access control in the sense that it ensures that only authorized subjects can access objects. Integrity addresses the preservation of information in that unauthorized or unwanted changes to objects are denied (and checked). Availability addresses the ability to obtain access within a reasonable amount of time upon request, in the sense that authorized requests for objects must be granted as quickly as system and network parameters allow. The term *access control* describes a broad range of controls, from forcing a user to provide a valid username and password to log on to preventing users from gaining access to a resource outside their sphere of access.

Access controls can be divided into the following seven categories of function or purpose. You should notice that some security mechanisms may acquire labels from multiple categories. Thus, for example, a fence can be both a preventive control and a deterrent control.

Preventive access control A preventive access control (sometimes called a preventative access control in CISSP materials) is deployed to stop unwanted or unauthorized activity from occurring. Examples of preventive access controls include fences, locks, biometrics, mantraps, lighting, alarm systems, separation of duties, job rotation, data classification, penetration testing, access control methods, encryption, auditing, presence of security cameras or closed circuit television (CCTV), smart cards, callback, security policies, security awareness training, and antivirus software.

Deterrent access control A deterrent access control is deployed to discourage violation of security policies. Deterrent controls pick up where prevention leaves off. A deterrent doesn't stop with trying to prevent an action; instead, it goes further to exact consequences in the event of an attempted or successful violation. Examples of deterrent access controls include locks, fences, security badges, security guards, mantraps, security cameras, trespass or intrusion alarms, separation of duties, work task procedures, awareness training, encryption, auditing, and firewalls.



Notice that *fences* (among others) are both preventive and deterrent access controls. This is true for many security items that appear in more than one category. For example, an 8-foot perimeter fence acts as a preventive access control by restricting open access and deters anyone without adequate means from scaling up and over it.

Detective access control A detective access control is deployed to discover unwanted or unauthorized activity. Often detective controls operate after the fact rather than in real time. Examples of detective access controls include security guards, guard dogs, motion detectors, recording and reviewing of events captured by security cameras or CCTV, job

rotation, mandatory vacations, audit trails, honeypots or honeynets, intrusion detection systems, violation reports, supervision and reviews of users, incident investigations, and intrusion prevention systems.

Corrective access control A corrective access control is deployed to restore systems to normal after an unwanted or unauthorized activity has occurred. Usually corrective controls are simple, such as terminating access or rebooting a system. Corrective controls have only minimal capability to respond to access violations. Examples of corrective access controls include intrusion detection systems, antivirus solutions, alarms, mantraps, business continuity planning, and security policies.

Recovery access control A recovery access control is deployed to repair or restore resources, functions, and capabilities after a violation of security policies. Recovery controls have more advanced or complex abilities to respond to access violations than corrective access controls. For example, recovery access can repair damage as well as prevent further damage. Examples of recovery access controls include backups and restores, fault-tolerant drive systems, server clustering, antivirus software, and database or virtual machine shadowing.

Compensation access control A compensation access control is deployed to provide various options to other existing controls to aid in enforcement and support of security policy. Examples of compensation access controls include security policy requirements or criteria, personnel supervision, monitoring, and work task procedures.

Compensation controls can also include controls used instead of more desirable or damaging controls. For example, if a guard dog cannot be deployed owing to proximity of a residential area, a motion detector with a spotlight and a barking sound playback device can be used instead.

Directive access control A directive access control is deployed to direct, confine, or control the actions of subjects to force or encourage compliance with security policies. Examples of directive access controls include security guards, guard dogs, security policy requirements or criteria, posted notifications, escape route exit signs, monitoring, supervision, work task procedures, and awareness training.

Access controls can be further categorized by how they are implemented. Where matters of implementation are concerned, the related categories are administrative, logical/technical, and physical:

Administrative access controls Administrative access controls are the policies and procedures defined by an organization's security policy to implement and enforce overall access control. Administrative access controls focus on two areas: personnel and business practices (for example, people and policies). Examples of administrative access controls include policies, procedures, hiring practices, background checks, data classification, security training, vacation history, reviews, work supervision, personnel controls, and testing.

Logical/technical access controls Logical access controls and technical access controls are the hardware or software mechanisms used to manage access to resources and systems and also provide protection for those resources and systems. Examples of logical or technical access controls include encryption, smart cards, passwords, biometrics, constrained interfaces, access control lists (ACLs), protocols, firewalls, routers, intrusion detection systems, and clipping levels.



We use the words *logical* and *technical* interchangeably within this concept.

Physical access controls Physical access controls are physical barriers deployed to prevent direct contact with systems or areas within a facility. Examples of physical access controls include guards, fences, motion detectors, locked doors, sealed windows, lights, cable protection, laptop locks, swipe cards, guard dogs, video cameras, mantraps, and alarms.

Access Control in a Layered Environment

No single access control mechanism is ever deployed on its own. In fact, combining various types of access controls is the only means by which a reasonably secure environment can be developed. Often multiple layers or levels of access controls are deployed to provide layered security, or defense in depth.

This idea is depicted using concentric circles of protection, which surround your assets and resources with logical circles of security protection. Thus, intruders or attackers need to overcome multiple layers of defense to reach protected assets. Layered security, or defense in depth, is considered a more logical approach to security than a traditional fortress mentality. In a fortress mentality, a single giant master wall is built around the assets, like the massive rock walls of a castle. The major flaw in such an approach is that large massive structures often have minor weakness and flaws; are difficult if not impossible to reconfigure, adjust, or move; and are easily seen and avoided by would-be attackers (in other words, they find easier ways into protected areas).

In a *layered security* (or *defense-in-depth*) deployment, your assets are surrounded by a layer of protection provided for by administrative access controls, which in turn is surrounded by a layer of protection consisting of logical or technical access controls, which is finally surrounded by a layer of protection that includes physical access controls. This concept of defense in depth highlights two important points. First, an organization's security policy ultimately provides the first or innermost layer of defense for your assets. Without a security policy, there is no real security that can be trusted. Security policies are one element of administrative access controls. Second, people are your last line of defense. People or personnel are the other focus for administrative access control. Only with proper training and education can your personnel implement, comply with, and support security elements defined in your security policy.

The Process of Accountability

One important purpose of security is to be able to hold people accountable for activities that their online personae (in other words, their user accounts) perform within the digital world of a computer network. The first step in this process is identifying the subject. In fact, several steps lead up to the ability to hold a person accountable for online actions: identification, authentication, authorization, auditing, and accountability.

Identification

Identification is the process by which a subject professes an identity and accountability is initiated. A user provides a username, a logon ID, a personal identification number (PIN), or a smart card to represent an identification process. Providing a process ID number also represents an identification process. Once a subject has identified itself, the claimed identity becomes accountable for any further actions undertaken by that subject. Information technology (IT) systems track activity by identities, not by subjects themselves. A computer doesn't know one human from another, but it does know that your user account is different from all other user accounts.

Authentication

Authentication is the process of verifying or testing that a claimed identity is valid. Authentication requires that a subject provide additional information that must correspond exactly to the identity professed. The most common form of authentication is a password, which is the first of three information factors (“something you know”) used for authentication:

Type 1 A Type 1 authentication factor is *something you know*. It is any string of characters you have memorized and can reproduce on a keyboard when prompted. Examples include a password, PIN, lock combination, passphrase, mother's maiden name, and so on.

Type 2 A Type 2 authentication factor is *something you have*. It is a physical device that you possess and must have on your person at the time of authentication. Examples include a smart card, token device, memory card, USB drive, and so on. This can also include your physical location, referred to as the *somewhere you are* factor.



The main difference between a memory card and a smart card is that a memory card is used only to store information, while a smart card has the ability to process data. We'll discuss these security methods in more detail in Chapter 19, “Physical Security Requirements.”

Type 3 A Type 3 authentication factor is *something you are*. It is a body part or a physical characteristic of your person. Examples of this factor include fingerprints, voice prints, retina patterns, iris patterns, face shapes, palm topology, hand geometry, and so on. This

factor is often labeled as a *biometric*, or a *biometric factor*. (We discuss these in more detail shortly.)

Each type of authentication factor is roughly the same in terms of the level of security provided in that only a single attack must succeed to overcome any single such factor. However, by number, each type is more secure than the one before it. For instance, a Type 3 factor is the most difficult to breach of the three just described. Nevertheless, a biometric factor may be overcome by creating a fake duplicate (like a gummi fingerprint). A Type 2 factor, the next most difficult to breach, can be overcome by physical theft, and a Type 1 factor can be overcome by a password attack. As you can see, a Type 3 factor is slightly more secure than a Type 2 factor, which is in turn more secure than a Type 1 factor.

These three basic factors (“something you know,” “something you have,” and “something you are”) are the most common elements in a fully functional security system. However, a few other factors also apply to the same security scenario in different ways, and with very different implications.

“Something” and “Somewhere”

In addition to these three commonly recognized factors, there are at least two others. One is *something you do*, such as writing a signature, typing a passphrase (keyboard dynamics), or speaking a phrase. Something you do is often included in the “something you are,” or Type 3, category.

Another factor, mentioned earlier, is *somewhere you are*, such as the computer terminal from which you log in or the phone number (identified by caller ID) or country (identified by your IP address) from whence you connect. Controlling access by physical location forces a subject to be present rather than connecting remotely. “Somewhere you are” is often included in the “something you have,” or Type 2, category.

Logical Location

Logical location can combine the ideas of “somewhere you are,” “something you have,” and “something you know.” A *logical location* access control restricts access based upon some form of logical identification, such as IP address, MAC address, client type, or protocol used. However, please note that logical location control should not be the only factor used because any type of address information can be spoofed using hacking tools.

Access can further be restricted to date and time of day or by transaction type. The former prevents access except within defined time periods. The latter is a content- or context-dependent control where access is dynamic based on transactions a subject wants to complete.

Multiple-Factor Authentication

Two-factor authentication occurs when two different factors are required to provide authentication. For example, when using a debit card at the grocery store, you must usually swipe the card (“something you have”) and enter a PIN (“something you know”) to complete the transaction. Strong authentication is simply any authentication that

requires two or more factors, but these are not necessarily factors of different types. As a general rule, when factors of different types are combined, the resultant authentication is more secure.

The concept behind two-factor authentication is that when two of the same factors are used together, the strength of the system is no greater than it would be if just one of the factors was used alone. More specifically, the same attack that could steal or obtain one instance of the factor could obtain all instances of the factor. For example, using two passwords together is no more secure than using a single password because a password-cracking attempt could discover both in a single successful attack. However, when two or more different factors are employed, two or more different types or methods of attack must succeed to collect all relevant authentication elements. For example, if a token, a password, and a biometric factor are all used for authentication, then a physical theft, a password crack, and a biometric duplication attack must all succeed simultaneously to gain entry into the system.

Once logon credentials of a proffered identity and its authentication factor(s) are supplied to a system, they are checked against a database of identities on that system. If an identity is located and correct authentication factor(s) supplied, the subject is authenticated.

Authorization

Once a subject is authenticated, its access must be *authorized*. The process of authorization ensures that the requested activity or object access is possible given the rights and privileges assigned to the authenticated identity (which we refer to as the *subject* from this point forward). Authorization indicates who is trusted to perform specific operations. In most cases, the system evaluates an access control matrix that compares the subject, the object, and the intended activity (we discuss the access control matrix in greater detail in Chapter 11, “Principles of Computer Design”). If the specific action is allowed, the subject is authorized; if disallowed, the subject is not authorized.

Keep in mind that just because a subject has been identified and authenticated, that does not automatically mean it has been authorized. It is possible for a subject to log onto a network (in other words, be identified and authenticated) yet be blocked from accessing a file or printing to a printer (in other words, by not being authorized to perform such activities). Most network users are authorized to perform only a limited number of activities on a specific collection of resources. Identification and authentication are “all-or-nothing” aspects of access control. Authorization occupies a wide range of variations between all and nothing for each individual subject or object within the environment. For example, a user may be able to read a file but not delete it. A user may be able to print a document but not alter the print queue. A user may be able to log onto a system but not be allowed to access any resources.

It is important to understand the differences between identification, authentication, and authorization. Although they are similar and are essential to all security mechanisms, they are distinct and must not be confused.

Auditing and Accountability

Auditing is the process by which online activities of user accounts and processes are tracked and recorded. Auditing produces audit trails. Audit trails can be used to reconstruct events and to verify whether a security policy or authorization was violated. When contents of audit trails are compared with authorization against authenticated user accounts, people associated with accounts can be held *accountable* for their online actions.

According to the National Institute of Standards and Technology (NIST) in its Minimum Security Requirements for Federal Information and Information Systems (FIPS 200), audit data recording must comply with the following requirements:

- Create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity.
- Ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.

An earlier standard promulgated by NIST (NISTIR5153) is now superseded by the preceding document, but it spells these requirements out in more (and more useful) detail:

- The system shall provide a mechanism for generating a security audit trail that contains information to support after-the-fact investigation of loss or impropriety and appropriate management response.
- The system shall provide end-to-end user accountability for all security-relevant events.
- The system shall protect the security audit trail from unauthorized access.
- The system shall provide a mechanism to dynamically control, during normal system operation, the types of events recorded.
- The system shall protect the audit control mechanisms from unauthorized access.
- The system shall, by default, cause a record to be written to the security audit trail for numerous specific security-related events.
- The system shall provide a privileged mechanism to enable or disable the recording of other events into the security audit trail.
- For each recorded event, the audit record shall identify several specific data points at a minimum.
- The character strings input as responses to password challenges shall not be recorded in the security audit trail.
- The audit control mechanism shall provide an option to enable or disable the recording of invalid user IDs during failed user authentication attempts.
- Audit control data (for example, audit event masks) shall survive system restarts.
- The system shall provide a mechanism for automatically copying security audit trail files to an alternative storage area after a customer-specifiable period of time.

- The system shall provide a mechanism for the automatic deletion of security audit trail files after a customer-specifiable period of time.
- The system shall allow site control of the procedure to be invoked when audit records are unable to be recorded.
- The system shall provide tools to monitor the activities (in other words, capture the keystrokes) of specific terminals or network connections in real time.



This list is based on the NISTIR 5153 document, but we have paraphrased only a small excerpt. To view all the details see document NISTIR 5153 at <http://csrc.nist.gov>. You can download the FIPS 200 document from <http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf>.

An organization's security policy can be properly enforced only if accountability is maintained. In other words, security can be maintained only if subjects are held accountable for their actions. Effective accountability relies on the capability to prove a subject's identity and track its activities. Thus, accountability builds on the concepts of identification, authentication, authorization, access control, and auditing.

Identification and Authentication Techniques

Identification is a fairly straightforward concept. A subject must provide an identity to a system to start the authentication, authorization, and accountability processes. Providing an identity might entail typing a username, swiping a smart card, waving a token device, speaking a phrase, or positioning your face, hand, or finger for a camera or scanning device. Without an identity, a system has no way to correlate an authentication factor with the subject. A subject's identity is typically considered to be public information.

Authentication verifies the identity of the subject by comparing one or more factors against a database of valid identities (in other words, user accounts). The authentication factor used to verify identity is typically considered to be private information. The ability of the subject and system to maintain the secrecy of the authentication factors for identities directly reflects the level of security of that system.

Identification and authentication always occur together as a single two-step process. Providing an identity is the first step, and providing the authentication factor(s) is the second step. Without both, a subject cannot gain access to a system—neither element alone is useful.

A subject can provide several types of authentication information (for example, “something you know,” “something you have,” and so on). Each authentication technique

or factor has its unique benefits and drawbacks. Thus, it is important to evaluate each mechanism in light of the environment in which it will be deployed to determine viability.

Passwords

The most common authentication technique is the use of *passwords*, but they are also considered the weakest form of protection. Passwords are poor security mechanisms for several reasons:

- Users typically choose passwords that are easy to remember and therefore easy to guess or crack.
- Randomly generated passwords are hard to remember; thus, many users write them down.
- Passwords are easily shared, written down, and forgotten.
- Passwords can be stolen through many means, including observation, recording and playback, and security database theft.
- Passwords are often transmitted in clear text or with easily broken encryption protocols.
- Password databases are often stored in publicly accessible online locations.
- Short passwords can be discovered quickly in brute-force attacks.

Password Selection

Passwords can be effective if selected intelligently and managed properly. There are two types of passwords: static and dynamic. *Static* passwords always remain the same. *Dynamic* passwords change after a specified interval of time or use. One-time passwords or single-use passwords are a variant of dynamic passwords that change every time they are used. One-time passwords are considered the strongest password type, at least in concept. Humans can't remember an infinite series of lengthy random character strings, which have only a single-attempt use before expiring. Thus, one-time passwords are often implemented with Type 2 factors using a processing device known as a *token* (we discuss tokens later in this chapter).

As the importance of maintaining security increases, so does the need to change passwords more frequently. The longer a password remains static and the more often the same password is used, the more likely it is that it will be compromised or discovered.

In some environments, initial passwords for user accounts are generated automatically. Often the generated password is a form of composition password. A *composition password* is a password constructed from two or more unrelated words joined together with a number or symbol in between. Composition passwords are easy for computers to generate, but they should not be used for extended periods of time because they are vulnerable to password-guessing attacks. If the algorithm for computer-generated passwords is discovered, all passwords created by the system are in jeopardy of being compromised.

A password mechanism that is slightly more effective than a basic password is a *passphrase*. A passphrase is a string of characters usually much longer than a password. Once a passphrase is entered, the system converts it into a virtual password for use by the

authentication process. Passphrases are often modified natural-language sentences to simplify memorization. Here's an example: "She \$\$\$ C shells ByE the c-shor." Using a passphrase has several benefits. It is difficult to crack a passphrase using a brute-force tool, and it encourages the use of a lengthy string with numerous characters yet is still easy to remember.

Another interesting password mechanism is the cognitive password. A cognitive password is usually a series of questions about facts or predefined responses that only a subject should know. For example, three to five questions such as these might be asked of the subject:

- What is your birth date?
- What is your mother's maiden name?
- What is the name of your division manager?
- What was your score on your last evaluation exam?
- Who was your favorite player in the 1984 World Series?

If all questions are answered correctly, the subject is authenticated. The most effective cognitive password systems ask a different set of questions each time. The primary limitation for cognitive password systems is that each question must be answered at the time of user enrollment (in other words, user account creation) and answered again during the logon process, which increases the time to complete that process. Cognitive passwords are often employed for phone- or web-based authentication at financial organizations, such as your bank. However, this type of password is considered to be inappropriate and insecure when it comes to protecting IT.

Many systems include password policies that restrict or dictate password characteristics. Common requirements include minimum length, minimum age, maximum age, requiring three or four character types (uppercase, lowercase, numbers, and symbols), and preventing password reuse. As needs for security increase, such restrictions should be tightened.

However, even with strong software-enforced password restrictions, it remains possible to create passwords that may be easily guessed or cracked. An organization's security policy must clearly define both the need for strong passwords and what a strong password is. Users need to be trained about security so they will respect an organization's security policy and adhere to its requirements. If end users create their own passwords, offer suggestions like the following to help them create strong ones:

- *Don't* reuse any part of your name, logon name, email address, employee number, Social Security number, phone number, extension, or other identifying name or code.
- *Don't* use dictionary words, slang, or industry acronyms.
- *Do* use nonstandard capitalization and spelling.
- *Do* switch letters and replace letters with numbers.

Password Security

When a malicious user or attacker seeks to obtain passwords, they can employ several methods, including network traffic analysis, password file access, brute-force attacks, dictionary attacks, and social engineering. *Network traffic analysis* (also known as

sniffing) is the process of capturing network traffic when a user is entering a password for authentication. Once a password is discovered, the attacker attempts to replay the packet containing the password against the network to gain access. If an attacker can gain access to the password database file, it can be copied and a password-cracking tool can be used against it to extract usernames and passwords.

Brute-force and dictionary attacks are types of password attacks that can be waged against a stolen password database file or a system's logon prompt. In a *dictionary attack*, the attacker uses a script of common passwords and dictionary words to attempt to discover an account's password. In a *brute-force attack*, a systematic trial of all possible character combinations is used to discover an account's password. Finally, a *hybrid attack* attempts a dictionary attack and then performs a type of brute-force attack. The follow-up brute-force attack is used to add prefix or suffix characters to passwords from the dictionary to discover one-upped-constructed passwords, two-upped-constructed passwords, and so on. A *one-upped-constructed* password is a password where a single character differs from its form in the dictionary. For example, "password1" is one-upped from "password," and so are "Password," "1password," and "passXword." This approach is often used to generate so-called rainbow tables, which map known passwords to equivalent hash values to speed password-cracking efforts. (Rainbow tables are discussed in more detail in Chapter 2, under the heading "Brute-Force and Dictionary Attacks.")

No matter what type of password attack is used, only read access is required for the password database. Write access is not required. Therefore, a wider number of user accounts can be employed to launch password-cracking attacks. From an intruder's perspective, this makes finding a weak user account more attractive than attacking an administrator or root account from the get-go to gain system access.

A *social-engineering attack* is an attempt by an attacker to obtain logon capabilities. It involves deceiving a user, sometimes over the telephone, into performing specific actions on a system, such as changing the password for an executive who is on the road or creating a user account for a new fictitious employee.

You can improve the security of passwords in several ways. *Account lockout* is a mechanism used to disable a user account after a specified number of failed logons. Account lockouts stop brute-force and dictionary attacks against a system's logon prompt. Once the logon attempt limit is reached, a message displaying the time, date, and location (in other words, the computer name or IP address) of the last successful or failed logon attempt appears. Users who suspect that their account is under attack or has been compromised can report this to a system administrator. Auditing can be configured to track logon success and failure. An intrusion detection system can easily identify logon prompt attacks and notify administrators.

Here are some other options to improve the security offered by password authentication:

- Use the strongest form of one-way encryption available for password storage.
- Never allow passwords to be transmitted over the network in clear text or with weak encryption.
- Use password verification tools and password-cracking tools against your own password database file. Require that weak or discovered passwords be changed.
- Disable idle user accounts for short periods of inactivity, such as a week or a month. Delete accounts no longer in use.

- Properly train users about the necessity of maintaining security and the use of strong passwords. Prohibit writing down or sharing passwords. Offer tips to prevent shoulder surfing or keyboard logging to capture passwords. Offer tips and techniques for creating strong passwords:
 - Require that users change passwords regularly. The more secure or sensitive the environment, the more frequently passwords should be changed.
 - Never display passwords in clear form on any screen or within any form. Instead, mask the display of the password at all times. This is a commonly recognized feature of software, such as displaying asterisks instead of letters when someone is typing a password into a logon dialog box.
 - Longer passwords, such as those with 16 characters or more, are harder for a brute-force password-cracking tool to discover. However, it's harder for people to remember longer passwords, which often leads users to write them down. Your organization should have a standard security awareness rule that no passwords should ever be written down. The only possible exception to that rule is that long, complex passwords for the most sensitive accounts, such as administrator or root, can be written down and stored in a vault or safety deposit box.
 - Create lists of passwords users should avoid. Easy-to-memorize passwords are often easily discovered by password-cracking tools.
 - If a root or administrator password is ever compromised, every password for every account should be changed. (In a high-security environment, a compromised system can never be fully trusted again. Thus, it may require formatting the drives and rebuilding the entire system from scratch.)
 - Hand out passwords in person after a user proves their identity. Never transmit passwords via email.

Biometrics

Another common authentication and identification technique is the use of *biometric factors*. Biometric factors fall into the Type 3, “something you are,” authentication category. A biometric factor is a behavioral or physiological characteristic that is unique to a subject. There are many types of biometric factors, including fingerprints, face scans, iris scans, retina scans, palm scans (also known as *palm topography* or *palm geography*), hand geometry, voice patterns, signature dynamics, and keystroke patterns (keystroke dynamics).

We now discuss biometric factors in more detail, taking into account the human body part they utilize and the information that each quantifies in order to make the most accurate identification possible:

Fingerprints The *macroscopic* (in other words, visible to the naked eye) patterns on the last phalange of fingers and thumbs make fingerprinting so effective for security. A type of fingerprinting known as *minutia matching* examines the microscopic view of the fingertips.

Unfortunately, minutia matching is affected by small changes to the finger, including temperature, pressure, and minor surface damage (such as sliding your fingers across a rough surface).

Face scans Face scans utilize the geometric patterns of faces for detection and recognition. They employ recognition technology known as *eigenfeatures* (facial metrics) or *eigenfaces*. (The German word *eigen* refers to recursive mathematics used to analyze intrinsic or unique numerical characteristics.)

Retina scans Retina scans focus on the pattern of blood vessels at the back of the eye. They are the most accurate form of biometric authentication (and are able to differentiate between identical twins), but also the least acceptable because retina scans can reveal medical conditions, such as high blood pressure and pregnancy. In addition, these types of scans often require a subject to use a cup reader that blows air directly into the eye.

Iris scans Focusing on the colored area around the pupil, iris scans are the second most accurate form of biometric authentication. However, iris scans cannot differentiate between identical twins. Iris scans are often recognized as having a longer useful authentication life span than other biometric factors. This is because the iris remains relatively unchanged throughout a person's life (barring eye damage or illness). Every other type of biometric factor is more likely to change over time. Iris scans are considered acceptable by general users because they don't involve direct contact with a reader and don't reveal personal medical information.

Palm scans Also known as *palm topography* or *palm geography*, palm scans utilize the whole area of the hand, including the palm and fingers. Palm scans function as a hand-sized fingerprint by analyzing the grooves, ridges, and creases as well as the fingerprints themselves.

Hand geometry Hand geometry recognizes the physical dimensions of the hand. This includes the width and length of the palm and fingers. This can be a mechanical or image-edge (in other words, visual silhouette) graphical solution.



Skin scans are not used as a form of biometric authentication because they cannot differentiate among all individuals.

Heart/pulse patterns This involves measuring the user's pulse or heartbeat to ensure that a real person is providing the biometric factor. This is often employed as a secondary biometric to support another type.

Voice pattern recognition This type of biometric authentication relies on the sound of a subject's speaking voice. This is different from speech recognition, which extracts communications from sound (in other words, automatic dictation software). Specifically, voice pattern recognition differentiates between one person's voice and another, while speech recognition differentiates between words within any person's voice.



Voice pattern recognition is often thought to have numerous benefits, such as its reliability and its function as a “natural” biometric factor. However, the idea of speech recognition is commonly confused with voice pattern recognition. Remember, voice pattern recognition differentiates between one person’s voice and another, while speech recognition differentiates between words within any person’s voice. The benefits of speech recognition include flexibility, hands- and eyes-free operation, reduced data entry time, elimination of spelling errors, and improved accuracy.

Signature dynamics This recognizes how a subject writes a string of characters. Signature dynamics examine how a subject performs the act of writing as well as features in a written sample. The success of signature dynamics relies upon pen pressure, stroke pattern, stroke length, and the points in time when the pen is lifted from the writing surface. However, the speed at which the written sample is created is usually not an important factor.

Keystroke patterns (keystroke dynamics) Keystroke patterns measure how a subject uses a keyboard by analyzing flight time and dwell time. *Flight time* is how long it takes between key presses, and *dwell time* is how long a key is pressed. Using keystroke patterns is inexpensive, nonintrusive, and often transparent to the user (for both use and enrollment). Unfortunately, using keystroke patterns for security is subject to wild variances. Simple changes in user behavior greatly affect this biometric, such as using only one hand, being cold, standing rather than sitting, changing keyboards, or sustaining an injury to the hand or a finger.

Biometric factors can be used as an identifying or authentication technique. Using a biometric factor instead of a username or account ID as an identification factor requires a one-to-many search of the offered biometric pattern against a stored database of enrolled and authorized patterns. As an identification technique, biometric factors are used in physical access controls. Using a biometric factor as an authentication technique requires a one-to-one match of the offered biometric pattern against a stored pattern for the offered subject identity. As an authentication technique, biometric factors are used in logical access controls.

The use of biometrics promises universally unique identification for every person on the planet. Unfortunately, biometric technology has yet to live up to this promise. For biometric factors to be useful, they must be extremely sensitive. The most important aspect of a biometric device is its accuracy. To use biometrics for identification, a biometric device must be able to detect minute differences in information, such as variations in the blood vessels in a person’s retina or tones and timbres in their voice. Because most people are basically similar, the level of detail required to authenticate a subject often results in false negative and false positive authentications.

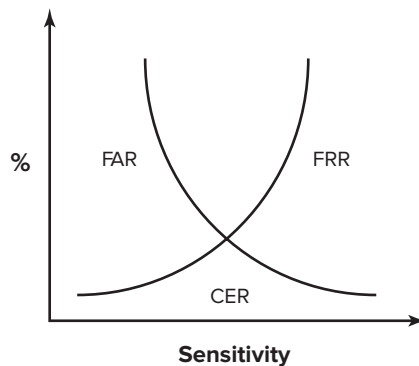
Biometric Factor Ratings

Biometric devices are rated for performance in producing false negative and false positive authentications. Most biometric devices have a sensitivity adjustment so they can be tuned to be more or less sensitive. When a biometric device is too sensitive, a Type 1 error occurs. A Type 1 error occurs when a valid subject is not authenticated. The ratio of Type 1 errors to valid authentications is known as the *false rejection rate (FRR)*. When a biometric device

is not sensitive enough, a Type 2 error occurs. A Type 2 error occurs when an invalid subject is authenticated. The ratio of Type 2 errors to valid authentications is called the *false acceptance rate (FAR)*.

The FRR and FAR are usually plotted on a graph that shows the level of sensitivity adjustment against the percentage of FRR and FAR errors (see Figure 1.1). The point at which the FRR and FAR are equal is known as the *crossover error rate (CER)* or the *equal error rate (ERR)*; these terms are used interchangeably. The CER level is used as a standard assessment point from which to measure the performance of a biometric device. The CER is used for a single purpose: to compare the accuracy of similar biometric devices (in other words, those focusing on the same biometric factor) from different vendors or different models from the same vendor. On the CER graph, the device with the lowest CER is overall the most accurate. In some situations, making a device more sensitive than the CER rate is preferable, such as on a metal detector at an airport.

FIGURE 1.1 Graph of FRR and FAR errors indicating the CER point



Biometric Registration

In addition to issues concerning the sensitivity of biometric devices, several other factors may make them less effective—namely, enrollment time, throughput rate, and acceptance. For a biometric device to work as an identification or authentication mechanism, subjects must be enrolled or registered. This means a subject's biometric factor must be sampled and stored in the device's database. The stored sample of a biometric factor is called a *reference profile* or a *reference template*.

The time required to scan and store a biometric factor varies greatly according to which physical or performance characteristic is measured. The longer it takes to enroll using a biometric mechanism, the less willingly the user community accepts the inconvenience. In general, enrollment times over 2 minutes are unacceptable. If you use a biometric characteristic that changes over time, such as a person's voice tones, facial hair, or signature pattern, re-enrollment must occur at regular intervals.

Once subjects are enrolled, the amount of time the system requires to scan and process them is the *throughput rate*. The more complex or detailed a biometric

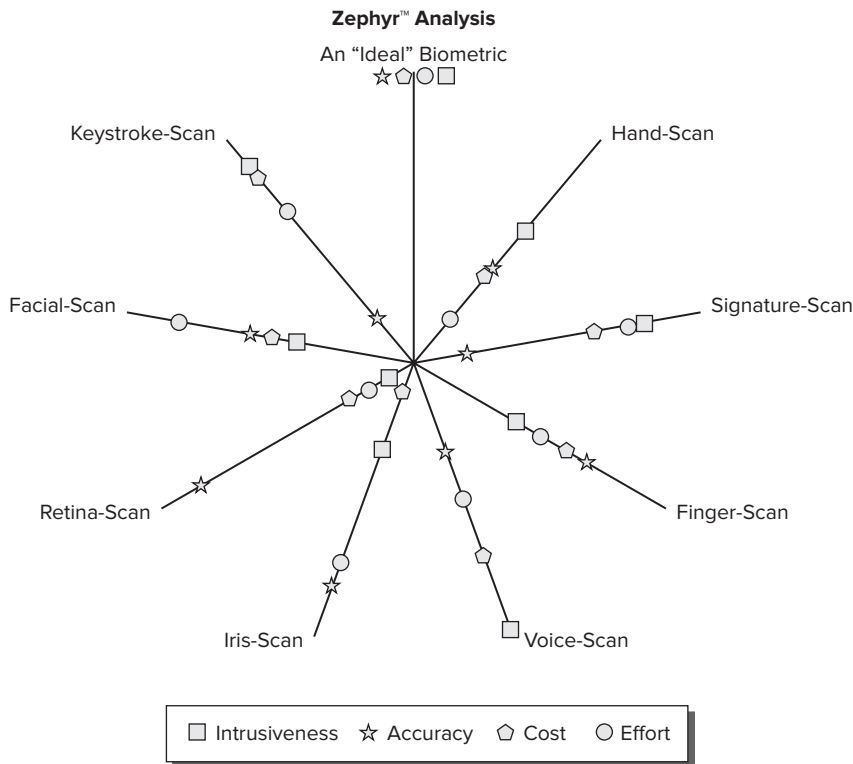
characteristic, the longer processing takes. Subjects typically accept a throughput rate of about six seconds or faster.

A subject's acceptance of a security mechanism depends upon many subjective perceptions, including privacy, invasiveness, and psychological or physical discomfort. Subjects may be concerned about transferring body fluids or may have health concerns about biometric-scanning devices.

Appropriate Biometric Usage

When selecting a biometric solution for a specific environment, you must consider numerous aspects. These aspects include which type of biometric factor is most suitable as well as the effectiveness and acceptability of the biometric factor. When comparing different types of biometric factors, a Zephyr chart is often used. A Zephyr chart rates various aspects, functions, or features of different biometric controls together on a single easy-to-read diagram (see Figure 1.2).

FIGURE 1.2 An example Zephyr chart



The *effectiveness* of access controls, specifically biometric controls, depends on how accurate one type of biometric factor is in comparison to others. Here is the common order of accuracy from most to least:

- Palm scan
- Hand geometry
- Iris scan
- Retina pattern
- Fingerprint
- Voice verification
- Facial recognition
- Signature dynamics
- Keystroke dynamics

The *acceptance* of biometrics is a rating of how well people accept use of specific biometric factors in their environment. An acceptance rating incorporates a person's view of how invasive and easy to use some specific biometric factor is and the level of health risk it presents. Here is the usual order of acceptance level from most to least:

- Iris scan
- Keystroke dynamics
- Signature dynamics
- Voice verification
- Facial recognition
- Fingerprint
- Palm scan
- Hand geometry
- Retina pattern



This list comes from work by A. K. Jain, a distinguished professor in the departments of computer science and engineering at Michigan State University. It is available through his Biometric Recognition website at <http://biometrics.cse.msu.edu>.

Tokens

Tokens (or *smart tokens*) are password-generating devices that subjects must carry with them. A token device is an example of a Type 2 factor, or “something you have.” A token can be a static password device, such as an ATM card or other memory card. To use an ATM card, you must supply the token (the ATM card itself) and your PIN. Tokens can also

be one-time or dynamic password devices that look like small calculators, or they might even be smart cards (to read more about smart cards, see Chapter 19). The device displays a string of characters (a password) for you to enter into the system.

There are four types of token devices:

- Static tokens
- Synchronous dynamic password tokens
- Asynchronous dynamic password tokens
- Challenge-response tokens

A *static token* can be a swipe card, a smart card, a floppy disk, a USB RAM dongle, or even something as simple as a key for a physical lock. Static tokens offer a physical means to prove identity. Static tokens often require an additional factor to provide authentication, such as a password or biometric factor. Most device static tokens host a cryptographic key, such as a private key, digital signature, or encrypted logon credentials. A cryptographic key can be used as an identifier or as an authentication mechanism. A cryptographic key is much stronger than a password because it is pre-encrypted using strong encryption, is significantly longer, and resides only in the token. Static tokens are used more as identification devices than as authentication factors.

A *synchronous dynamic password token* generates passwords at fixed time intervals. Time interval tokens require synchronizing the clock on an authentication server with the clock on the token device. The subject enters a generated password into the system along with a PIN, passphrase, or password. This generated password provides identification, and the PIN/password provides authentication.

An *asynchronous dynamic password token* generates passwords based on occurrence of some event. An event token requires the subject to press a key on the token and on the authentication server. This advances to the next password value. The generated password and the subject's PIN, passphrase, or password are entered into the system for authentication.

One-Time Password Generators

As we discussed earlier, one-time passwords are dynamic passwords that change every time they are used. They can be effective for security purposes, except that humans can rarely remember passwords that change so frequently. *One-time password generators* create passwords for your users and make one-time passwords reasonable to deploy. Users need a token device (in other words, a password generator), must understand the logon procedure, and often need to memorize a short PIN, depending on which generator they use. With device-based authentication systems, an environment can benefit from the strength of one-time passwords without relying on users' memorization skills.

The five widely recognized one-time password generator systems are synchronous, PIN synchronous, asynchronous, PIN asynchronous, and transaction synchronous. Systems that use a PIN require entry of an additional memorized key sequence to complete the authentication process.

Challenge-response tokens generate passwords or responses based on instructions from the authentication system. The authentication system displays a challenge, usually a code or a passphrase. This challenge is entered into the token device. The token responds to the challenge, and its response is keyed into the system for authentication.

Using token authentication systems offers much stronger security than using password authentication alone. Token systems use two or more factors to establish identity and provide authentication. In addition to knowing the username, password, PIN, code, and so on, the subject must be in physical possession of the token device.

However, token systems do have failings. If the battery dies or the device breaks, the subject is unable to gain access. Token devices can get lost or stolen. Tokens must be stored and managed intelligently because if a token system is compromised, it can be difficult and expensive to replace. Furthermore, human factors can render tokens less secure than they are designed to be. First and foremost, if a user writes the access code or PIN on the token device, the security of the token system is compromised. Users must recognize that loaning a token and PIN, even to a co-worker, violates security.

Tickets

Ticket authentication is a mechanism that employs a third-party entity to prove identification and provide authentication. The most common and well-known ticket system is Kerberos. Kerberos was developed under Project Athena at MIT. We'll discuss Kerberos and its tickets later in this chapter.



The Kerberos name is borrowed from Greek mythology. A three-headed dog named Kerberos guards the gates to the underworld, but in the myth, the three-headed dog faces inward, preventing escape rather than denying entrance.

Single Sign-On

Single sign-on (SSO) is a mechanism that allows a subject to be authenticated only once on a system yet remain able to access resource after resource unhindered by repeated authentication prompts. With SSO, once a subject is authenticated, it can roam the network freely and access resources and services without further authentication challenges.

The primary disadvantage to SSO is that once an account is compromised, a malicious subject gains unrestricted access. In other words, a maximum level of unauthorized access is gained simply through password disclosure. SSO typically supports stronger passwords because a subject must memorize only a single password. Furthermore, SSO eases administration by reducing the number of locations on which an account must be defined for the subject. You can enable SSO through authentication systems or through scripts that provide logon credentials automatically when prompted.

Kerberos, SESAME, KryptoKnight, NetSP, thin clients, directory services, and scripted access are examples of SSO mechanisms. Two or more SSO mechanisms can be combined into a single security solution. It is typical for Kerberos to be combined with another SSO mechanism. For example, under Windows Server 2008 (as well as Windows Server 2003), it is possible to employ the native directory service (Active Directory), which is integrated with Kerberos and other SSO options, including thin clients (in other words, Terminal Services) and scripted access (in other words, logon scripts).

Kerberos

Kerberos is a trusted third-party authentication protocol that can be used to provide a single sign-on solution and to provide protection for logon credentials. Kerberos relies upon symmetric-key cryptography (also known as private-key cryptography), specifically, Advanced Encryption Standard (AES), and it provides end-to-end security for authentication traffic between the client and the key distribution center (KDC). Kerberos provides confidentiality and integrity for authentication traffic.

The Kerberos authentication mechanism centers on a trusted server (or servers) that hosts the functions of the KDC, a ticket-granting service (TGS), and an authentication service (AS). Generally, the Kerberos central server that hosts all these services is simply referred to as the KDC. Kerberos uses symmetric-key cryptography to authenticate clients to servers. All clients and servers are registered with the KDC, so it maintains the secret keys for all network members.

A complicated exchange of tickets (in other words, cryptographic messages) between clients, network servers, and the KDC is used to prove identity and provide authentication. This allows a client to request resources from the server with full assurance that both client and server are who they claim to be. An exchange of encrypted tickets also ensures that no logon credentials, session keys, or authentication messages are ever transmitted in clear text.

Kerberos tickets have specific lifetimes and usage parameters. Once a ticket expires, a client must request a renewal or a new ticket to continue communications with any server.

The Kerberos logon process is as follows:

1. The user types a username and password into the client.
2. The client encrypts the credentials with AES for transmission to the KDC.
3. The KDC verifies the user credentials.
4. The KDC generates a TGT by hashing the user's password.
5. The TGT is encrypted with AES for transmission to the client.
6. The client installs the TGT for use until it expires.

The Kerberos server or service access process is as follows:

1. The client sends its TGT back to the KDC with a request for access to a server or service.
2. The KDC verifies the ongoing validity of the TGT and checks its access control matrix to verify that the user has sufficient privilege to access the requested resource.
3. A service ticket (ST) is generated and sent to the client.

4. The client sends the ST to the server or service host.
5. The server or service host verifies the validity of the ST with the KDC.
6. Once identity and authorization is verified, Kerberos activity is complete. The server or service host then opens a session with the client and begins communications or data transmission.

Limitations of Kerberos

Kerberos is a versatile authentication mechanism that works over local LANs, local logons, remote access, and client-server resource requests. However, Kerberos presents a single point of failure—the KDC. If the KDC is compromised, the secret key for every system on the network is also compromised. Also, if a KDC goes offline, no subject authentication can occur.

Kerberos has other limitations or problems:

- Dictionary and brute-force attacks on the initial KDC response to a client may reveal a subject's password. In fact, direct password-guessing attacks can be waged against a KDC unimpeded. A countermeasure to such attacks is to deploy a preauthentication service to check logon credentials and watch for access attacks before granting a subject access to the KDC.
- Issued tickets are stored in memory on the client and server.
- Malicious subjects can replay captured tickets if they are reused within their lifetime window.
- Issued tickets, specifically the TGT, are based on a hash of the user's password with an added time stamp for expiration.
- Kerberos encrypts only authentication traffic (in other words, mechanisms for proving identity); it does not provide security for subsequent communication sessions or data transmissions.

Other Examples of Single Sign-On

Although Kerberos may be the most widely recognized (and deployed) form of single sign-on, it is not the only one of its kind. In this section, we summarize other SSO mechanisms you may encounter.

The Secure European System for Applications in a Multivendor Environment (SESAME) is a system developed to address weaknesses in Kerberos. However, it did not compensate for all problems with Kerberos completely. Eventually later Kerberos versions and various vendor implementations resolved its initial problems. In the professional security world, SESAME is no longer considered a viable product.

KryptoKnight is a peer-to-peer authentication solution developed by IBM. It was incorporated into the NetSP product. Like SESAME, KryptoKnight and NetSP never took off and are no longer widely used.

Thin clients are low-end client systems that connect over a network to a server system. Thin clients originated in the mainframe world where host-terminal connections enabled

dumb terminals to interact with and control centralized mainframes. These terminals had no processing or storage abilities. The idea of thin clients has been replicated on modern client-server environments using interface software applications that act as clients to server-hosted environments. All processing and storage takes place on the server, while the client provides an interface for the subject through a local keyboard, mouse, and monitor. Some thin clients are also called *remote control tools*, used for remote desktop access and remote assistance, or “on the fly” remote connectivity tools such as DameWare.

A *directory service* is a centralized database of objects that includes information about resources available to a network along with information about subjects such as users and computers. It can be understood as a telephone directory for network services and assets. Users, clients, and processes consult the directory service to learn where a desired system or resource resides. Then once this address or location is known, access can be directed toward it. A directory service must be authenticated to before queries and lookup activities can be performed. Even after authentication, the directory service will reveal only certain information to a subject based on that subject’s assigned privileges. Directory services are often based on the Lightweight Directory Access Protocol (LDAP). Some well-known directory services include Microsoft’s Active Directory and Novell’s NetWare Directory Services (NDS), now known as eDirectory.

Scripted access or *logon scripts* are used to establish communication links by providing an automated process by which logon credentials are transmitted to resource hosts at the start of a logon session. Scripted access can often simulate SSO even though the environment still requires a unique authentication process to connect to each server or resource. Scripts can be used to implement SSO in those environments where true SSO technologies are not available. However, scripts and batch files should be stored in a protected area because they usually contain access credentials.

Access Control Techniques

Once a subject has been identified and authenticated and accountability has been established, it must be authorized to access resources or perform actions. Authorization can occur only after a subject’s identity has been verified through authentication. Systems provide authorization through the use of access controls. Access controls manage the type and extent of access subjects have to objects. There are two primary categories for access control techniques: discretionary and nondiscretionary. Nondiscretionary can be further subdivided into specific techniques, such as mandatory, role-based, and task-based access controls.

There are several forms of access controls that define how subjects access and interact with objects in a variety of ways. Each system has its own security properties that individually distinguish and differentiate it from all others. The types of access control systems are described in the following sections.

Discretionary Access Controls

A system that employs *discretionary access controls (DACs)* allows the owner or creator of an object to control and define subject access to that object. That is, access control is based on the discretion (in other words, a decision) of the owner. Access is granted or denied in a discretionary environment based on the identity of the subject (which is typically the user account name). For example, if a user creates a new spreadsheet file, that user is the owner of that file. As the owner of the file, that user can modify the permissions on that file to grant or deny access to other subjects.

DACs are often implemented using access control lists on objects. Each ACL defines the types of access granted or restricted to individual or grouped subjects. Discretionary access control does not offer a centrally controlled management system because owners can alter the ACLs on their objects at will. Thus, access is more dynamic than it is with mandatory access controls.

DAC environments can be extended beyond just controlling the type of access between subjects and objects via ACLs by including or applying time controls, transaction controls, and other forms of ID-focused controls (in other words, device, host, protocol, address, and so on). Within a DAC environment, users' privileges can be suspended while they are on vacation, resumed when they return, or terminated when they leave an organization.



The U.S. government labels access controls that do not rely upon policy to define access as discretionary; however, corporate environments and nongovernmental organizations will often label such environments as *need to know*.

Nondiscretionary Access Controls

Nondiscretionary access controls are used in a rule-based system in which a set of rules, restrictions, or filters determines what can and cannot occur on the system, such as granting subject access, performing an action on an object, or accessing a resource. Access is not based on administrator or owner discretion and does not focus on user identity. (Thus, nondiscretionary access control is the opposite of discretionary in much the same way as non-A is the opposite of A.) Rather, a static set of rules that governs the whole environment is used to manage access (in other words, nondiscretionary access implies a centrally controlled management system).

In general, rule-based access control systems are more appropriate for environments that experience frequent changes to data permissions (in other words, changing the security domain or label for objects). This is because rule-based systems can implement sweeping changes just by changing centralized rules without having to manipulate or “touch” every subject and/or object in the environment. However, in most cases, once rules are established, they remain fairly static and unchanged throughout the life of the environment.

In rule-based access control systems, control is based on a specific profile created for each user. A common example of such a system is a firewall. A firewall is governed by a set of rules or filters defined by the administrator. Users are able to communicate across the firewall because they have initiated transactions that are allowed by the defined rules. Users are able to accomplish this because they have client environments configured to permit some transactions and to deny all others; these are their specific profiles. A formal definition of a rule-based access control (or specifically, a *rule-based security policy*) is found in RFC 2828, “Internet Security Glossary.” This document includes the following definition for the term *rule-based security policy*: “A security policy based on global rules imposed for all users. These rules usually rely on comparison of the sensitivity of the resource being accessed and the possession of corresponding attributes of users, a group of users, or entities acting on behalf of users.”

Mandatory Access Controls

Mandatory access controls rely upon the use of classification labels. Each classification label represents a security domain or a realm of security. A *security domain* is a realm of common trust that is governed by a specific security policy for that domain. Subjects are labeled by their level of clearance (which is a form of privilege). Objects are labeled by their level of classification or sensitivity. For example, the military uses the labels of top secret, secret, confidential, sensitive but unclassified (SBU), and unclassified (see Chapter 5, “Security Management Concepts and Principles”).



Despite the title just cited for Chapter 5 in this book, the corresponding domain in the CISSP common body of knowledge is now known as “Information Security Governance and Risk Management,” and it stresses the key role that risk assessment and management, and best practices in security governance, play in managing security of all kinds.

In a mandatory access control system, subjects are able to access objects that have the same or a lower level of classification. An expansion of this access control method is known as *need to know*. Subjects with higher clearance levels are granted access to highly sensitive resources only if their work tasks require such access. If they don’t have a need to know, even if they have sufficient clearance, they are denied access.

Mandatory access control (MAC) is prohibitive rather than permissive. If access is not specifically granted, it is forbidden. MAC is generally recognized as being more secure than DAC but neither as flexible nor as scalable. The relative scale for security is evident in the ISO Standard Common Criteria for Computer Security (ISO 15408), a standard used to permit users to specify security requirements, vendors to specify security attributes for products, and evaluators to determine if such products embody claimed attributes. The Common Criteria, often abbreviated as CC, includes evaluation criteria and specifies mandatory protection as a higher level of security than

discretionary protection (for more information about CC, see Chapter 12, “Principles of Security Models”).

Using security labels in mandatory access controls presents some interesting problems. First, for a mandatory access control system to function, every subject and object must have a security label. Depending on the environment, security labels can refer to sensitivity, value to the organization, need for confidentiality, classification, department, project, and so on. The military security labels mentioned earlier range from highest sensitivity to lowest: top secret, secret, confidential, sensitive but unclassified (SBU), and unclassified. Common corporate or commercial security labels are confidential, proprietary, private, sensitive, and public. Security classifications indicate a hierarchy of sensitivity, but each level is distinct.

Classifications within a MAC environment are of three types:

Hierarchical environments Hierarchical environments relate various classification labels in an ordered structure from low security to medium security to high security. Each level or classification label in the structure is related. Clearance in one level grants the subject access to objects in that level as well as to all objects in all lower levels but prohibits access to all objects in higher levels.

Compartmentalized environments In compartmentalized environments, there is no relationship between one security domain and another. To gain access to an object, the subject must have specific clearance for its security domain.

Hybrid environments A hybrid environment combines both hierarchical and compartmentalized concepts so that each hierarchical level may contain numerous subdivisions that are isolated from the rest of the security domain. A subject must not only have the correct clearance but also the need to know for a specific compartment to gain access to the compartmentalized object. Possessing the need to know for one compartment within a security domain does not grant a subject access to any other compartment. Each compartment has its own unique and specific need to know. If you have a need to know (based on your assigned work tasks), then you are granted access. If you don't have a need to know, your access is blocked. A hybrid MAC environment provides more granular control over access but becomes increasingly difficult to manage as the size of the environment (in other words, number of classifications, objects, and subjects) increases.

Role-Based Access Control

Systems that employ role-based or task-based access controls define a subject's ability to access an object via subject roles (in other words, job descriptions) or tasks (in other words, work functions). If a subject occupies a management position, it will have greater access to resources than a subject who is in a temporary job. Role-based access controls are useful in volatile environments with frequent personnel changes because access depends on a job description (in other words, a role or task) rather than on subject identity.

Role-based access control (RBAC) and groups within a DAC environment may serve a similar purpose, but they differ in their deployment and use. They are similar in that both serve as containers to collect users into manageable units. However, a user can belong to more than one group. In addition to collecting rights and permissions from each group, individual user accounts may also be directly assigned rights and permissions.

In a DAC system, even with groups, access is still based on the discretion of an owner and focuses control on the identity of the user. When an RBAC system is employed, a user may have only a single role, but users may also be assigned multiple roles. Users have only the rights and permissions that belong to such roles, and there are no additional individually assigned rights or permissions. Furthermore, access is not determined by owner discretion; it derives from the inherent responsibilities of an assigned role (in other words, job description). Also, access focuses on the assigned role, not on user identity. Two different users with the same assigned role will have the same access and privileges.

RBAC is becoming increasingly attractive to corporate entities with high rates of employee turnover. It also allows company-specific security policies to be directly mapped and enforced to map directly into an organization's hierarchy and management structure. This implies that roles or job descriptions within an RBAC system are often hierarchical, meaning that roles are related in a low-to-high fashion so that higher roles are created by adding access and privileges to lower ones. Often, RBAC solutions can replace MAC and DAC environments.

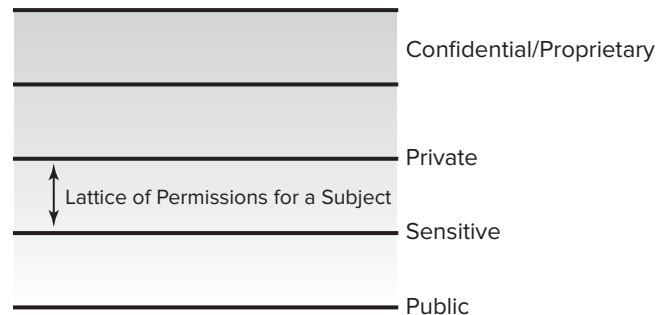
Another method related to RBAC is called *task-based access control* (TBAC). TBAC is basically the same as RBAC, but instead of being assigned a single role, each user is assigned an array of tasks. These items all relate to assigned work tasks for the person associated with a user account. Under TBAC, access remains based on rules (in other words, on work tasks) and focuses on controlling access by tasks assigned rather than by user identity.

Lattice-Based Access Controls

Some, if not most, nondiscretionary access controls can be labeled as *lattice-based access controls*. Lattice-based access controls define upper and lower bounds of access for every relationship between a subject and an object. These boundaries can be arbitrary, but they usually follow military or corporate security label levels.

A subject with the lattice permissions shown in Figure 1.3 can access resources up to private and down to sensitive but cannot access confidential, proprietary, or public resources. Subjects under lattice-based access controls acquire a *least upper bound* and a *greatest lower bound* of access to labeled objects based on their assigned lattice positions. Lattice-based access controls were originally developed to address information flow, which primarily concerns itself with confidentiality. A common example of a lattice-based access control is a mandatory access control.

FIGURE 1.3 A representation of the boundaries provided by lattice-based access controls



Access Control Methodologies and Implementation

There are two primary access control methodologies: centralized and decentralized (or distributed). *Centralized* access control implies that all authorization verification is performed by a single entity within a system. *Decentralized* access control, or *distributed* access control, implies that various entities located throughout a system perform authorization verification.

Centralized and Decentralized Access Control

Centralized and decentralized access control methodologies offer the same benefits and drawbacks found in any centralized or decentralized system. A small team or individual can manage centralized access control. Administrative overhead is lower because all changes are made in a single location and a single change affects the entire system. However, centralized access control also presents a single point of failure. If system elements are unable to access the centralized access control system, subjects and objects cannot interact. Two examples of centralized access control are Remote Authentication Dial-In User Service (RADIUS) and Terminal Access Controller Access Control System (TACACS).

Decentralized access control often requires several teams or multiple individuals. Administrative overhead is higher because changes must be implemented across numerous locations. Maintaining homogeneity across a system becomes more difficult as the number of access control points increases. Changes made to any individual access control point affect only aspects of the systems that rely upon that specific access control point. Decentralized

access control has no single point of failure. If an access control point fails, other access control points may be able to balance the load until the control point is repaired; in addition, objects and subjects that don't rely upon the failed access control point can continue to interact normally. Domains and trusts are commonly used in decentralized access control systems.

A domain is a realm of trust or a collection of subjects and objects that share a common security policy. Each domain's access control operates independently from those of other domains. This results in decentralized access control when multiple domains are involved. To share resources from one domain to another, a trust is established. A trust is simply a security bridge established between two domains that allows users from one domain to access resources in another. Trusts can be one way only, or they can be two way.

RADIUS and TACACS

RADIUS centralizes authentication for remote dial-up connections. A network that employs a RADIUS server is configured so the remote access server passes dial-up user logon credentials to the RADIUS server for authentication. This process is similar to the process used by domain clients sending logon credentials to a domain controller for authentication. Using an authentication server such as RADIUS or TACACS that is separate from the primary remote access server system provides the benefit of keeping auditing and access settings on a system other than the remote access server, thus providing greater security. RADIUS and other remote authentication protocols and services are designed to transport authentication, authorization, and session configuration information between a remote access server (aka a *network access server*) and a centralized authentication server (often known as a *domain controller*). Note also that a single RADIUS or TACACS server can support many remote access servers and acts as a central clearinghouse for such servers.

RADIUS was defined in RFC 2138 and is now covered in RFC 2865. It is used primarily to provide an additional layer of protection against intrusions via dial-up connections. RADIUS supports dynamic passwords and callback security. It acts as a proxy for remote clients because it acts on behalf of clients to obtain authentication on the network. RADIUS acts as a client for the network by requesting authentication in much the same way that a typical client would. Likewise, within the RADIUS architecture, a remote access server is configured as a RADIUS client.

Owing to the success of RADIUS, an enhanced version of RADIUS named DIAMETER was developed; it is designed to support all forms of remote connectivity, not just dial-up. However, RADIUS and DIAMETER are not interoperable. Eventually, DIAMETER's features were added back into RADIUS. Today, only a version of RADIUS that supports all types of remote access is available.

TACACS is an alternative to RADIUS. TACACS is available in three versions: original TACACS, extended TACACS (XTACACS), and TACACS+. TACACS integrates the authentication and authorization processes. XTACACS keeps the authentication, authorization, and accounting processes separate. TACACS+ improves XTACACS by adding two-factor authentication. TACACS and RADIUS operate similarly, and TACACS

provides the same functionality as RADIUS. However, RADIUS is based on an Internet standard, whereas TACACS is more proprietary (although widely used). TACACS is defined in RFC 1492.

These forms of centralized access control, specific to remote access, provide an additional layer of security for a private network. They prevent LAN authentication systems and domain controllers from being attacked directly by remote attackers. When a separate system for remote access users is deployed, even if that system is compromised, only remote access users are affected; the rest of the LAN still functions unhindered.

Access Control Administration

Access control administration is the collection of tasks and duties assigned to an administrator to manage user accounts, access, and accountability. A system's security is based on effective administration of access controls. Remember that access controls rely upon four principles: identification, authentication, authorization, and accountability. As they relate to access control administration, these principles transform into three main responsibilities:

- User account management
- Activity tracking
- Access rights and permissions management

Account Administration

User account management involves creating, maintaining, and closing user accounts. Although these activities may seem mundane, they are essential to a system's access control capabilities. Without properly defined and maintained user accounts, a system is unable to establish identity, perform authentication, provide authorization, or track accountability.

Creating New Accounts

Creating new user accounts is a simple process, but it must be protected and secured via organizational security policy procedures. User accounts should not be created at an administrator's whim or in response to some particular request. Rather, account management should follow a stringent procedure that flows from the HR department's hiring or promotion procedures.

The HR department should make a formal request for a user account for a new employee. That request should include the classification or security level to be assigned to the new employee's user account. The new employee's department manager and the

organization's security administrator should verify that security assignment. Once the request is verified, only then should a new user account be created. Creating user accounts outside established security policies and procedures creates holes and oversights that can be exploited by malicious subjects. A similar process for increasing or decreasing an existing user account's security level is also required.

As part of the hiring process, new employees should be trained on organization security policies and procedures. Before hiring is complete, employees must sign an agreement committing to uphold the organization's security standards. Many organizations opt to craft a document that stipulates that violating security policy is grounds for dismissal as well as grounds for prosecution under federal, state, and local laws. When passing on the user account ID and temporary password to a new employee, organizations should conduct a review of the password policy and acceptable use restrictions at that time.

The initial creation of a new user account is often called an *enrollment*. The enrollment process creates a new identity and establishes the factors the system needs to perform authentication. It is critical that the enrollment process be completed fully and accurately. It is also critical that the identity of the individual being enrolled be proved through whatever means your organization deems necessary and sufficient. Photo ID, birth certificate, background check, credit check, security clearance verification, FBI database search, and even calling references are all valid forms of verifying a person's identity before enrolling them in any secured system.

Account Maintenance

Throughout the life of a user account, ongoing maintenance is required. Organizations with fairly static organizational hierarchies and low employee turnover or promotion will conduct significantly less account administration than an organization with a flexible or dynamic organizational hierarchy and high employee turnover and promotion rates. Most account maintenance deals with altering rights and privileges. Procedures similar to those used when creating new accounts should be established to govern how access is changed throughout the life of a user account. Unauthorized increases or decreases in an account's access capabilities can cause serious security repercussions.

When employees leave an organization, their user accounts should be disabled, deleted, or revoked. Whenever possible, this task should be automated and tied into the HR department. In most cases, when someone's paychecks are stopped, that person should no longer have logon capabilities. Temporary or short-term employees should have specific expiration dates programmed into their user accounts. This maintains a degree of control established at the time of account creation without requiring ongoing administrative oversight.

Account, Log, and Journal Monitoring

Activity auditing, account tracking, and system monitoring are also important aspects of access control management. Without these capabilities, it is impossible to hold subjects

accountable. Through the establishment of identity, authentication, and authorization, tracking the activities of subjects (including how many times they access objects) offers direct and specific accountability. We discuss auditing and monitoring as an aspect of operations security and as an essential element in a secure environment in Chapter 14, “Auditing and Monitoring.”

User accounts, event logs, and system journals help piece together the state of affairs for a server at any referenced point along the timeline of its operation. Event logs and system journals capture events, changes, messages, and other data that describe what activities occurred on a system. Thus, they are commonly used to support conclusions drawn about any incidents that might warrant investigation. When an account is obtained after an outside attacker exploits a vulnerable service, you can bet the server documented some aspects of that incident in its event logs and system journals.

Access Rights and Permissions

Assigning access to objects is an important part of implementing an organizational security policy. Not all subjects should be granted access to all objects. Not all subjects should have the same functional capabilities on objects. A few specific subjects should access only some objects; likewise, certain functions should be accessible only to a few specific subjects.

For instance, the data entry department in any given organization does not require explicit access to the resources and information found in the accounting department. Therefore, not all subjects (those in data entry) require access to particular objects (in this case, accounting). Only managers within the accounting department may access financial data, and only supervisors are responsible for creating and maintaining that data.

The Principle of Least Privilege

The *principle of least privilege* arises from the complex structure that results when subjects are granted access to objects. This principle states that subjects should be granted only as much access to objects as is required to accomplish their assigned work tasks. The principle has a converse that should be followed as well: Subjects should be blocked from accessing objects that are not required by their work tasks. The principle of least privilege is most often linked with DAC, but this concept applies to all types of access control environments, including non-DAC, MAC, RBAC, and TBAC.



We utilize acronyms throughout this book to conserve space and to make terms easier to memorize. On the exam, you will be tested with all terms and acronyms spelled out, so there will be no confusion between a rule-based access control (RBAC) system and a role-based access control (RBAC) system. Study each system and its defining characteristics carefully.

Keep in mind that the idea of privilege usually means the ability to write, create, alter, or delete data. Thus, limiting and controlling privilege based upon this concept can be a protection mechanism for data integrity. If users can change only those data files that their work tasks require them to change, then the integrity of all other files in the environment is protected.

This principle relies on the assumption that all users have a definite and distinct job description that is well defined and understood. Without a specific job description, it is not possible to know what privileges a user does or does not need.

Need-to-Know Access

A related principle in the realm of mandatory access control environments is known as *need to know*. Within a specific classification level or security domain, some assets or resources may be sectioned off or compartmentalized. Such resources are restricted from general access even to subjects with otherwise sufficient clearance. Compartmentalized resources require an additional level of formalized access approval before they can be used by subjects. Subjects are granted access when they can justify their work-task-related reason for access or their need to know. Often, need to know is determined by a domain supervisor and is granted only for a limited period of time.

Determining which subjects have access to which objects is a function of the organizational security policy, the organizational hierarchy of personnel, and the implementation of an access control model. Thus, the criteria for establishing or defining access can be based on identity, roles, rules, classifications, location, time, interfaces, need to know, and so on. Access control models are formal descriptions of a *security policy*, which is a document that encapsulates the security requirements for an organization and prescribes the steps necessary to achieve the desired security. Access control models (or security models) are used in security evaluations and assessments as well as in tools used to validate security.

Excessive Privilege and Creeping Privileges

It's important to guard against two problems related to access control: excessive privilege and creeping privileges. *Excessive privilege* is when a user has more access, privilege, or permission than assigned work tasks dictate. If a user account is discovered to have excessive privilege, additional and unnecessary privileges should be immediately revoked. *Creeping privileges* involve a user account accumulating privileges over time as job roles and assigned tasks change. This can occur because new tasks are added to a user's job and the related or necessary privileges are added as well but no privileges are ever removed, even if a related work task is no longer associated with or assigned to that user. Creeping privileges result in excessive privilege. You can prevent both of these issues by applying the principle of least privilege properly and stringently.

Users, Owners, and Custodians

When discussing access to objects, three subject labels are used: user, owner, and custodian. A *user* is any subject who accesses objects on a system to perform some action or accomplish a work task. An *owner*, or information owner, is the person who has final corporate responsibility for classifying and labeling objects and protecting and storing data. The owner may be liable for negligence if they fail to perform due diligence in establishing and enforcing security policies to protect and sustain sensitive data. A *custodian* is a subject who has been assigned or delegated the day-to-day responsibility of properly storing and protecting objects.

A user is any end user on the system. The owner is typically the CEO, president, or department head. The custodian is typically the IT staff or the system security administrator.

Separation of Duties and Responsibilities

The separation of duties and responsibilities is a common practice that prevents any single subject from being able to circumvent or disable security mechanisms. When core administration or high-authority responsibilities are divided among several subjects, no one subject has sufficient access to perform significant malicious activities or to bypass imposed security controls.

A separation of duties creates a checks-and-balances system where multiple subjects verify each other's actions and must work in concert to accomplish necessary work tasks. Separating duties makes perpetration of malicious, fraudulent, or otherwise unauthorized activities much more difficult and broadens the scope of detection and reporting. It is easy for individuals to perform unauthorized acts if they think they can get away with it. Once two or more people are involved, the committal of an unauthorized activity requires that each person agrees to keep a shared secret. This typically serves as a significant deterrent rather than as a means to corrupt a group en masse. The separation of duties can be static or dynamic. The static separation of duties is accomplished by assigning privileges based on written policies that don't change often. The dynamic separation of duties is used when security requirements cannot be determined until the system is active and functioning.

An example of a properly enforced separation of duties is to prevent the security administrator from being able to access system administration utilities or to perform changes to system configuration not related to security. For example, a security administrator needs no more than read access to system logs. In this manner, a separation of duties helps prevent conflicts of interest in the types of privileges assigned to administrators as well as users in general. Figure 1.4 illustrates common privileges that should not be combined with others in order to properly enforce a separation of duties.

FIGURE 1.4 A segregation of duties control matrix

	Control Group	Systems Analyst	Application Programmer	Help Desk and Support Mgr.	End User	Data Entry	Computer Operator	DB Administrator	Network Administrator	System Administrator	Security Administrator	Tape Librarian	Systems Programmer	Quality Assurance
Control Group		X	X	X		X	X	X	X	X			X	
Systems Analyst	X			X	X		X				X	X		
Application Programmer	X			X	X	X	X	X	X	X	X	X	X	
Help Desk and Support Mgr.	X	X	X		X	X		X	X	X		X	X	
End User		X	X	X			X	X	X			X	X	X
Data Entry	X		X	X			X	X	X	X	X		X	
Computer Operator	X	X	X		X	X		X	X	X	X		X	
DB Administrator	X		X	X	X	X	X		X	X			X	
Network Administrator	X		X	X	X	X	X	X				X		
System Administrator	X		X	X		X	X	X				X		
Security Administrator		X	X			X	X					X	X	
Tape Librarian		X	X	X	X				X	X	X		X	
Systems Programmer	X		X	X	X	X	X	X			X	X		X
Quality Assurance					X							X	X	

X—Combination of these functions may create a potential control weakness.

© 2005 Information Systems Audit and Control Association (ISACA). All rights reserved. Used with permission.



The segregation of duties control matrix is not an industry standard; rather, it's a guideline that indicates which positions should be separated and require compensating controls if combined. This matrix illustrates potential segregation of duties and should not be viewed or used as an absolute mandate; instead, use it to help identify potential conflicts so proper questions may be asked to identify compensating controls.

Summary

The first domain of the CISSP CBK is Access Control. Access controls are central to establishing a secure system. They rely upon identification, authentication, authorization, and accountability. Access control is the management, administration, and implementation of granting or restricting subject access to objects.

The first step in access control is verifying the identities of subjects on the system, commonly known as authentication. Other methods are available to authenticate subjects, including passwords and passphrases, biometric scans, tokens, and tickets.

Once a subject is authenticated, their access must be managed (authorization) and their activities logged so ultimately the person can be held accountable for the user account's online actions. Again, this is why we believe accountability and access control are mutually dependent, equally important components of a much larger and reliable security framework.

There are various models for access control or authorization. These include discretionary and nondiscretionary access controls. There are at least three important subdivisions of nondiscretionary access control: mandatory, role-based, and task-based access control.

Access can be managed for an entire network at once. Such systems are known as single sign-on solutions. Remote access clients pose unique challenges to LAN security and often require specialized tools such as RADIUS or TACACS.

Access control administration represents the collection of tasks and duties assigned to an administrator as they relate to managing user accounts, access, and accountability. This includes user account management, activity tracking, and access rights and permissions management, all of which are subject to life cycle considerations related to their creation, ongoing maintenance, and deletion or removal at the end of their useful lives.

Account, log, and journal monitoring also play an important role in managing access control because they provide the mechanisms and the data necessary to hold subjects accountable for their actions.

Assigned access to objects is a key aspect of implementing organizational security policy. This is where the principle of least privilege comes into play; it dictates that subjects should only obtain as much access as is required to accomplish assigned work tasks. This also explains need to know as a mandatory access control mechanisms designed to restrict access to information only to those whose job responsibilities require them to possess that information.

When access to objects is under discussion, three key subject labels are often used—namely, user, owner, and custodian. The user is a subject who accesses objects on a system pursuant to performing an action or accomplishing some work task. The owner is the subject responsible for classifying and labeling objects and protecting and storing data. A custodian is a subject to whom responsibility for properly storing and protecting objects is assigned or delegated.

Separation of duties provides a necessary set of checks and balances whereby multiple subjects must verify each other's actions on objects and work together to accomplish necessary work tasks. Separation of duties helps to reduce the possibility (and the perpetration) of malicious, fraudulent, and other unauthorized uses of objects. Proper separation and segregation of duties ensures that no single individual obtains sufficient access to violate security policy without involving other individuals.

Exam Essentials

Know the various types of access control. Access controls may be preventive (to stop unwanted or unauthorized activity from occurring), deterrent (to discourage violation of security policy), detective (to discover unwanted or unauthorized activity), corrective (to restore systems to normal after an unwanted or unauthorized activity has occurred), recovery (to repair or restore resource, functions and capabilities after a violation of security policy has occurred), compensation (to provide various options to other existing controls to aid in enforcement and support of security policy), directive (to direct, confine, or control the action of subjects to force or encourage compliance with security policy), administrative (policies or procedures to implement and enforce overall access control), logical/technical (hardware or software mechanisms used to manage access to resources and systems and to provide protection for those resources and systems), and physical (physical barrier deployed to prevent direct contact with systems or areas within a facility).

Know the common access control techniques. Common access control techniques include discretionary, mandatory, nondiscretionary, rule based, role based, and lattice based. Access controls are used to manage the type and extent of access subjects have to objects. This is important to system security because such controls define who has access to what.

Understand access control administration. Securely creating new user accounts, managing and maintaining user accounts on an ongoing basis, auditing/logging/monitoring subject activity, and assigning and managing subject access are important aspects of keeping a system secure. Security is an ongoing task, and administration is how you keep a system secure over time.

Know details about each of the access control models. There are two primary categories of access control techniques: discretionary and nondiscretionary. Nondiscretionary can be further subdivided into specific techniques, such as mandatory, role-based, and task-based access control.

Understand the processes of identification and common identification factors. The processes of identification include subject identity claims by using a username, user ID, PIN, smart card, biometric factors, and so on. They are important because identification is the first step in authenticating a subject's identity and proper access rights to objects.

Understand the processes of authentication and the various authentication factors.

Authentication involves verifying the authentication factor provided by a subject against the authentication factor stored for the claimed identity, which could include passwords, biometrics, tokens, tickets, SSO, and so on. In other words, the authentication process ensures that a subject is who they claim to be and grants object rights accordingly.

Understand the processes of authorization. Authorization ensures that the requested activity or object access is possible given the rights and privileges assigned to the authenticated identity. This is important because it maintains security by providing proper access rights for subjects.

Understand the strengths and weaknesses of passwords. Users typically choose passwords that are easy to remember and therefore easy to guess or crack, which is one weakness often associated with passwords. Another is that randomly generated passwords are hard to remember, and thus many users write them down. Passwords are easily shared and can be stolen through many means. Additionally, passwords are often transmitted in clear text or with easily broken encryption protocols, and password databases are often stored in publicly accessible online locations. Finally, short passwords can be discovered quickly in brute-force attacks. On the other hand, passwords can be effective if selected intelligently and managed properly. It is important to change passwords frequently; the more often the same password is used, the more likely it will be compromised or discovered.

Know the two access control methodologies and implementation examples. Access control methodologies include centralized access control, in which authorization verification is performed by a single entity within a system, and decentralized access control, in which authorization verification is performed by various entities located throughout a system. Remote authentication mechanisms such as RADIUS and TACACS are implementation examples; they are used to centralize the authentication of remote dial-up connections.

Understand the use of biometrics. Biometric factors are used for identification or authentication. FRR, FAR, and CER are important aspects of biometric devices. Fingerprints, face scans, iris scans, retina scans, palm topography, palm geography, voice pattern, signature dynamics, and keystroke patterns are often used along with other authentication factors, such as a password, to provide an additional method to control authentication of subjects.

Understand single sign-on. Single sign-on (SSO) is a mechanism that allows a subject to be authenticated only once on a system and be able to access resource after resource unhindered by further authentication prompts. Kerberos, SESAME, KryptoKnight, NetSP, thin clients, directory services, and scripted access are all SSO mechanisms.

Understand access control administration. Access control administration breaks down into three areas of administrative responsibility: user account management, activity tracking, and access rights and permission management. It's important to understand the tasks and activities related to each of these three areas and how they can impact security.

Appreciate how account, log, and journal monitoring enforce accountability. Managing access control also means holding subjects accountable for their actions. Account, log, and journal monitoring and auditing tools provide the means whereby accountability may be assigned to specific subjects.

Understand key concepts involved in assigning object access. The principle of least privilege dictates that subjects be granted no more permissions than they absolutely need to perform their assigned work duties. Need to know means the subjects should only be granted object access when they require such access to do their jobs (and adds a specific dimension to individual objects that general security classification schemes cannot provide). In general, the default access rule for objects should be to deny access unless specific subjects require access to do their jobs (and then such access should permit only those actions that the job entails and no more).

Be able to explain these subject labels: user, owner, and custodian. A user is a subject who accesses objects in the course of performing some action or accomplishing a work task. The owner is the subject responsible for classifying and labeling objects and for protecting and storing data on any system. A custodian is a subject to whom the protect and store role for some object or collection of objects has been delegated or assigned.

Understand why separation or segregation of duties is important. When subjects have permissions that enable them to conduct entire transactions, change general security settings, or alter policy, they have the ability to transgress against policy without necessarily setting off alarms or alerts about potential or actual policy violations. By design, separation or segregation of duties breaks up permissions and access necessary to make such sweeping changes across multiple job roles so that no single individual should be able to undertake such activities.

Written Lab

1. Name at least seven access control types.
2. Describe the three primary authentication factor types.
3. Identify at least three access control techniques.
4. What is the principle of least privilege?

Answers to Written Lab

1. Access control types include preventive access control, deterrent access control, detective access control, corrective access control, recovery access control, compensation access control, directive access control, administrative access control, logical or technical access control, and physical control.
2. A Type 1 authentication factor is “something you know.” A Type 2 authentication factor is “something you have.” A Type 3 authentication factor is “something you are.”
3. Discretionary access controls, nondiscretionary or rule-based access controls, mandatory access controls, role-based access controls, and lattice-based access controls.
4. The principle of least privilege defines the access permissions that are granted to a given user to achieve some specified tasks. It is the security concept and best practice of allowing only the necessary permissions to achieve such tasks.

Review Questions

1. What is access?
 - A. Functions of an object
 - B. Information flow from objects to subjects
 - C. Unrestricted admittance of subjects on a system
 - D. Administration of ACLs
2. Which of the following is true?
 - A. A subject is always a user account.
 - B. The subject is always the entity that provides or hosts the information or data.
 - C. The subject is always the entity that receives information about or data from the object.
 - D. A single entity can never change roles between subject and object.
3. Which of the following types of access control uses fences, security policies, security awareness training, and antivirus software to stop an unwanted or unauthorized activity from occurring?
 - A. Preventive
 - B. Detective
 - C. Corrective
 - D. Authoritative
4. _____ access controls are the hardware or software mechanisms used to manage access to resources and systems and to provide protection for those resources and systems.
 - A. Administrative
 - B. Logical/technical
 - C. Physical
 - D. Preventive
5. What is the first step of access control?
 - A. Accountability logging
 - B. ACL verification
 - C. Subject authorization
 - D. Subject identification

6. _____ is the process of verifying or testing the validity of a claimed identity.
- A. Identification
 - B. Authentication
 - C. Authorization
 - D. Accountability
7. Which of the following is an example of a Type 2 authentication factor?
- A. “Something you have,” such as a smart card, ATM card, token device, and memory card
 - B. “Something you are,” such as fingerprints, voice print, retina pattern, iris pattern, face shape, palm topology, and hand geometry
 - C. “Something you do,” such as type a passphrase, sign your name, and speak a sentence
 - D. “Something you know,” such as a password, personal identification number (PIN), lock combination, passphrase, mother’s maiden name, and favorite color
8. Which of the following is not a reason why using passwords alone is a poor security mechanism?
- A. When possible, users choose easy-to-remember passwords that are easy to guess or crack.
 - B. Randomly generated passwords are hard to remember, and thus many users write them down.
 - C. Short passwords can be discovered quickly in brute-force attacks only when used against a stolen password database file.
 - D. Passwords can be stolen through many means, including observation, recording and playback, and security database theft.
9. Which of the following is not a valid means to improve the security offered by password authentication?
- A. Enabling account lockout controls
 - B. Enforcing a reasonable password policy
 - C. Using password verification tools and password-cracking tools against your password database file
 - D. Allowing users to reuse the same password
10. What can be used as an authentication factor that is a behavioral or physiological characteristic unique to a subject?
- A. Account ID
 - B. Biometric factor
 - C. Token
 - D. IQ

11. What does the crossover error rate (CER) for a biometric device indicate?
 - A. The sensitivity is tuned too high.
 - B. The sensitivity is tuned too low.
 - C. The false rejection rate and the false acceptance rate are equal.
 - D. The biometric device is not properly configured.

12. Which of the following is not an example of an SSO mechanism?
 - A. Kerberos
 - B. KryptoKnight
 - C. TACACS
 - D. SESAME

13. _____ access controls rely upon the use of labels.
 - A. Discretionary
 - B. Role-based
 - C. Mandatory
 - D. Nondiscretionary

14. A network environment that uses discretionary access controls is vulnerable to which of the following?
 - A. SYN flood
 - B. Impersonation
 - C. Denial of service
 - D. Birthday attack

15. What is the most important aspect of a biometric device?
 - A. Accuracy
 - B. Acceptability
 - C. Enrollment time
 - D. Invasiveness

16. Which of the following is not an example of a deterrent access control?
 - A. Encryption
 - B. Auditing
 - C. Awareness training
 - D. Antivirus software

17. Kerberos provides the security services of _____ protection for authentication traffic.
- A. Availability and nonrepudiation
 - B. Confidentiality and authentication
 - C. Confidentiality and integrity
 - D. Availability and authorization
18. Which of the following forms of authentication provides the strongest security?
- A. Password and a PIN
 - B. One-time password
 - C. Passphrase and a smart card
 - D. Fingerprint
19. Which of the following is the least acceptable form of biometric device?
- A. Iris scan
 - B. Retina scan
 - C. Fingerprint
 - D. Facial geometry
20. Why is separation of duties important for security purposes?
- A. It ensures that multiple people can do the same job.
 - B. It prevents an organization from losing important information when they lose important people.
 - C. It prevents any single security subject (person) from being able to make major security changes without involving other subjects.
 - D. It helps subjects concentrate their talents where they will be most useful.

Answers to Review Questions

1. B. Access is the transfer of information from an object to a subject.
2. C. The subject is always the entity that receives information about or data from the object. The subject is also the entity that alters information about or data stored within the object. The object is always the entity that provides or hosts information or data. A subject can be a user, a program, a process, a file, a computer, a database, and so on. The roles of subject and object can switch while two entities, such as a program and a database or a process and a file, communicate to accomplish a task.
3. A. A preventive access control is deployed to stop an unwanted or unauthorized activity from occurring. Examples of preventive access controls include fences, security policies, security awareness training, and antivirus software.
4. B. Logical/technical access controls are the hardware or software mechanisms used to manage access to resources and systems and to provide protection for those resources and systems. Examples of logical or technical access controls include encryption, smart cards, passwords, biometrics, constrained interfaces, access control lists, protocols, firewalls, routers, intrusion detection systems, and clipping levels.
5. D. Access controls govern subjects' access to objects. The first step in this process is identifying who the subject is. In fact, there are several steps preceding actual object access: identification, authentication, authorization, and accountability.
6. B. Authentication is the process of verifying or testing the validity of a claimed identity.
7. A. A Type 2 authentication factor is "something you have." This could be a smart card, ATM card, token device, or memory card.
8. C. Brute-force attacks can be used against password database files and system logon prompts.
9. D. Preventing password reuse increases security by preventing the theft of older password database files, which can be used against the current user passwords.
10. B. A biometric factor is a behavioral or physiological characteristic that is unique to a subject, such as fingerprints and face scans.
11. C. The point at which the FRR and FAR are equal is the crossover error rate (CER). The CER level is used as a standard assessment point from which to measure the performance of a biometric device.
12. C. Kerberos, SESAME, and KryptoKnight are examples of SSO mechanisms. TACACS is a centralized authentication service used for remote access clients.
13. C. Mandatory access controls rely on use of labels. A system that employs discretionary access controls allows the owner or creator of an object to control and define subject access to that object. Nondiscretionary access controls are also called *role-based access controls*. Systems that employ nondiscretionary access controls define a subject's ability to access an object through the use of subject roles or tasks.

14. B. A discretionary access control environment controls access based on user identity. If a user account is compromised and another person uses that account, they are impersonating the real owner of the account.
15. A. The most important aspect of a biometric factor is its accuracy. If a biometric factor is not accurate, it may allow unauthorized users into a system.
16. D. Antivirus software is not a deterrent access control, though it is regarded as an access control that has recovery, corrective, and preventative characteristics.
17. C. Kerberos provides confidentiality and integrity protection security services for authentication traffic.
18. C. Among these options, passphrase and a smart card provide the strongest authentication security because they deliver two-factor authentication.
19. B. Of the options listed, retina scan is the least accepted biometric device because it blows air into the eye and can reveal personal health issues.
20. C. Of the options listed, separation or segregation of duties is intended to make fraud, theft, or malicious violations of security policy more difficult by involving multiple subjects (people) in making security changes or conducting complete transactions with security or monetary significance.

